II. RELATED APPEALS, INTERFERENCES, AND JUDICIAL PROCEEDINGS

Applicant respectfully notes that the following co-pending applications are or have been previously on appeal before the Board, which may have a bearing on the Board's decision in this appeal: 1) Application No. 09/880,632 ("the '632 Application"); 2) Application No. 10/146,988 ("the '988 Application); and 3) Application No. 10/146,967 ("the '967 Application").

An appeal was filed for the '632 Application with an Appeal Brief being filed August 20, 2007. An Examiner's Answer was mailed December 13, 2007. As of this time, no decision has been rendered by the Board for the appeal of the '632 Application.

An appeal was filed for the '967 Application with an Appeal Brief being filed February 17, 2009. An Examiner's Answer was mailed April 29, 2009. As of this time, no decision has been rendered by the Board for the appeal of the '967 Application.

An appeal was filed for the '988 Application with an Appeal Brief being filed June 9, 2006. An Examiner's Answer was mailed April 18, 2007, and a Reply Brief was filed June 18, 2007. As of this time, no decision has been rendered by the Board for the appeal of the '988 Application.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

The following provides a concise explanation of the subject matter defined in each of the independent claims involved in the appeal, referring to the specification by page and line number and to the drawings by reference characters, as required by 37 C.F.R. § 41.37(c)(1)(v). Each element of the claims is identified by a corresponding reference to the specification and drawings where applicable. Note that the citation to passages in the specification and drawings for each claim element does not imply that the limitations from the specification and drawings should be read into the corresponding claim element.

Independent claim 1 recites in a communication network, a method of TCP state migration comprising the steps of:

> a) establishing a TCP/IP communication session between a client computer (*e.g.*, client 410 of Fig. 4) and a first server computer (*e.g.*, server 450 of Fig. 4), said first server computer part of a plurality of server computers forming a web cluster containing information (*e.g.*, web cluster 490 of Fig. 4), said communication session established for the transfer of data contained within said information (*see* p. 22, line 22-p. 23, line 19, of Specification);
>
> b) handing off said communication session to a selected server computer (*e.g.*, server 452 of Fig. 4, and *see* p. 23, lines 9-13 of the Specification) from said first server computer over a persistent control channel using TCP handoff modules (*e.g.*, Upper TCP module 522 and Bottom TCP module 524 of Fig. 5C, and *see* p. 8, line 25-p. 11, line 6, and p. 29, line 27-p. 30, line 6 of the Specification) that are dynamically loadable (*see* p. 17, line 24-p. 18, line 15 of the Specification) within TCP/IP stacks in operating systems located at both said first server computer and said selected server computer, that implement a TCP handoff protocol that works within kernel levels of an existing TCP/IP protocol (*see* p. 23, line 21-p. 26, line 29 of the Specification); and
>
> c) migrating a first TCP state of said first server computer to said selected server computer, and a second TCP state of said selected server computer to said first server computer over said control channel (*e.g.*, p. 10, line 26-p. 11, line 28 of the Specification).

Independent claim 11 recites in a communication network, a method of TCP state migration comprising the steps of:

a) establishing a TCP/IP communication session between a client computer (*e.g.*, client 410 of Fig. 4) and a first server computer (*e.g.*, server 450 of Fig. 4), said first server computer part of a plurality of server computers forming a web cluster containing information (*e.g.*, web cluster 490 of Fig. 4), said communication session established for the transfer of data contained within said information;

b) monitoring traffic associated with establishing said TCP/IP communication session to understand a first initial TCP state of said first server computer associated with said TCP/IP communication session, at a first bottom-TCP (BTCP) module at said first server computer (Bottom TCP module 524 of Fig. 5 and BTCP module 830 of Fig. 8, and *see* p. 9, lines 16-20 and p. 11, lines 8-11 of the Specification);

c) receiving a web request associated with said TCP/IP communication session at said first BTCP module at said first server computer (block 910 of Fig. 9 and block 1310 of Fig. 13, and *see* p. 10, lines 7-8 of the Specification);

d) examining content of said web request (*see* p. 10, lines 8-10);

e) determining which of said plurality of server computers, a selected server computer, can best process said web request, based on said content (block 930 of Fig. 9, and *see* p. 10, lines 13-16 of the Specification);

f) handing off said communication session to said selected server computer (*e.g.*, server 452 of Fig. 4) from said first server computer over a persistent control channel, if said selected server computer is not said first server computer (*see* p. 10, line 26-p. 11, line 28, and p. 23, lines 9-13 of the Specification);

g) monitoring traffic associated with handing off said TCP/IP communication session to understand a second initial TCP state of said selected server computer associated with said TCP/IP communication session, at a second BTCP module at said selected server computer (*e.g.*, BTCP module 870 of Fig. 8, and *see* p. 12, lines 1-13 of the Specification);

h) migrating said first initial TCP state to said selected server computer over said control channel, such that said second BTCP module can calculate a first TCP state for said first server computer in said TCP/IP

communication session (*e.g.*, p. 12, line 15-p. 13, line 8 of the Specification);

i) sending a second initial TCP state of said selected server computer to said first BTCP module (*e.g.*, BTCP module 830 of Fig. 8), such that said first BTCP module can calculate a second TCP state for said selected server computer in said TCP/IP communication session;

j) forwarding data packets received at said first BTCP module from said client to said selected server computer, by changing said data packets to reflect said second TCP state and a second IP address of said selected server computer (*e.g.*, block 1320 of Fig. 13, and *see* p. 12, line 15-p. 13, line 8 of the Specification);

k) sending response packets from said selected server computer directly to said client computer (*see* Figs. 3 and 4) by changing said response packets to reflect said first TCP state and a first IP address of said first server computer (*e.g.*, block 1440 of Fig. 14, and *see* p. 12, line 15-p. 13, line 8 of the Specification);

l) terminating said TCP/IP communication session at said first server computer when said TCP/IP communication session is closed (*e.g.*, p. 13, lines 10-19).

Independent claim 26 recites a server computer comprising:

an upper TCP (UTCP) module (*e.g.*, UTCP module 522 of Fig. 5C, and UTCP modules 810 and 850 of Fig. 8) located above a TCP module (*e.g.*, TCP module 520 of Figs. 5B and 5C, and TCP modules 820 and 860 in Fig. 8) in an operating system of said server computer;

a bottom TCP (BTCP) module (*e.g.*, BTCP module 524 of Fig. 5C, and BTCP modules 830 and 870 of Fig. 8) located below said TCP module, said UTCP, TCP, and BTCP modules implementing a method of handing off a communication session between a first node (*e.g.*, server 450 of Fig. 4, and "front-end" node of Fig. 8) and second node (*e.g.*, server 452 of Fig. 4, and "back-end" node of Fig. 8) in a cluster network (*e.g.*, cluster 490 of Fig. 4) that works within the kernel level of an existing TCP/IP protocol, by migrating TCP states associated with said first and second nodes (*see* p. 10, line 26-p. 12, line 13 of the Specification).

Dependent claim 2 recites the method as described in Claim 1, wherein said step a) comprises the steps of:

receiving a SYN packet from said client at a first BTCP module located at said first server computer (Fig. 10:1010; Spec., p. 33, ln. 9-12);

sending said SYN packet upstream to a first TCP module located above said first BTCP module in a first operating system of said first server computer (Fig. 10:1020; Spec., p. 33, ln. 13-14);

receiving a first SYN/ACK packet from said first TCP module (Fig. 10:1030; Spec., p. 33, ln. 16-18);

parsing said first initial TCP state from said first SYN/ACK packet, including a first initial sequence number for said first TCP module associated with said TCP/IP communication session (Fig. 10:1040; Spec., p. 33, ln. 18-20);

sending said SYN/ACK packet to said client (Fig. 10:1050; Spec., p. 33, ln. 20-22);

receiving an ACK packet from said client at said first BTCP module (Fig. 10:1060; Spec., p. 33, ln. 24-25);

sending said ACK packet to said first TCP module (Fig. 10:1070; Spec, p. 33, ln. 25-26);

receiving a web request packet associated with said TCP/IP communication session at said first BTCP module at said first server computer (Fig. 10:1080; Spec., p. 33, ln. 26-29),

storing said SYN, ACK and said web request packet at said first server computer (Fig. 10:1090; Spec., p. 34, ln. 1-5).

Dependent claim 3 recites the method as described in Claim 2, wherein said step b) comprises the steps of:

examining content of said web request packet (Spec., p. 34, ln. 18-20);

determining which of said plurality of server computers, a selected server computer, can best process said WEB request packet, based on said content (Spec., p. 10, ln. 7-16);

sending a handoff request from said first BTCP module to a second BTCP module at said selected server computer over said control channel, if said selected server computer is not said first server computer (Spec., p. 35, ln. 7-14);

including said SYN packet and said ACK packet in said handoff request packet (Spec., p. 35, ln. 16-18);

changing a first destination IP address of said SYN packet to a second IP address of said selected server computer, at said second BTCP module (Spec., p. 37, ln. 5-8);

sending said SYN packet to said second TCP module (Spec., p. 37, ln. 8-10);

receiving a second SYN/ACK packet at said second BTCP module;

parsing said second initial TCP state from said second SYN/ACK packet, including a second initial sequence number, for said second TCP module, that is associated with said TCP/IP communication session (Spec., p. 37, ln. 12-17);

changing a second destination IP address of said ACK packet to said second IP address, at said second BTCP module (Spec., p. 37, ln. 5-8);

updating said ACK packet to reflect said second TCP state of said selected server computer in said communication session (Spec., p. 37, ln. 19-22);

sending said ACK packet that is updated to said second TCP module (Spec., p. 37, ln. 24-26); and

sending a handoff acknowledgement message to said first BTCP module (Spec., p. 37, ln. 27 – p. 38, ln. 2).

Dependent claim 4 recites the method as described in Claim 3, wherein step c) comprises the steps of:

> monitoring traffic associated with establishing said TCP/IP communication session in step a), at said first BTCP module, to parse a first initial TCP state of said first server computer, said first initial TCP state associated with said TCP/IP communication session (Spec., p. 11, ln. 8 – p. 12, ln. 13); and
>
> migrating said first initial TCP state to said second BTCP module over said control channel by including said first initial TCP state in said handoff request packet, said first initial TCP state including a first sequence number, such that said second BTCP module can calculate said first TCP state for said first server computer in said TCP/IP communication session (Spec., p. 11, ln. 8 – p. 12, ln. 13).

Dependent claim 5 recites the method as described in Claim 3, wherein step c) comprises the steps of:

> monitoring traffic associated with handing off said TCP/IP communication session at said second BTCP module, to parse a second initial TCP state of said selected server computer, said second initial TCP state associated with said TCP/IP communication session (Spec., p. 11, ln. 8 – p. 12, ln. 13); and
>
> migrating said second initial TCP state of said selected server computer to said first BTCP module by including said second initial TCP state in said handoff acknowledgment packet, said second initial TCP state including a second initial sequence number, such that said first BTCP module can calculate said second TCP state for said selected server computer in said TCP/IP communications session (Spec., p. 11, ln. 8 – p. 12, ln. 13).

Dependent claim 6 recites the method as described in Claim 2, comprising the further steps of:

> intercepting a connection indication message sent from said first TCP module to an application layer above said first TCP module at a first upper- TCP (UTCP) module, said connection indication message sent by said first TCP module upon establishing said communication session (Spec. p. 10, ln. 2-5; p. 34, ln. 10-11); and
>
> holding said connection indication message at said first UTCP module (Spec., p. 10, ln. 2-5; p. 34, ln. 10-11).

Dependent claim 7 recites the method as described in Claim 6, wherein said method comprises the further steps of:

> sending a reset packet from said first BTCP module upon receiving said handoff acknowledgment packet to said first TCP module (Spec., p. 35, ln. 22-25);
>
> discarding said connection indication message at said first UTCP module (Spec., p. 36, ln. 1-4);
>
> receiving incoming data packets from said client at said first BTCP module (Spec., p. 38, ln. 19-21);
>
> changing said destination addresses of said incoming data packets to said second IP address (Spec., p. 38, ln. 25-27);
>
> updating sequence numbers and TCP checksum in said data packets to reflect said second TCP state of said selected server computer (Spec., p. 38, ln. 27-29); and
>
> forwarding said data packets to said selected server computer (Spec., p. 38, ln. 29 – p. 39, ln. 2).

Dependent claim 8 recites the method as described in Claim 6, comprising the further steps of:

> sending notification from said first BTCP module to said first UTCP module to release said connection indication message, if said selected server computer is said first server computer (Spec., p. 40, ln. 15-20);
>
> sending incoming data packets, including said web request packet, from said client, received at said first BTCP module, upstream (Spec., p. 40, ln. 20-26).

Dependent claim 9 recites the method as described in Claim 1, comprising the further step of:

intercepting outgoing response packets from said selected server computer at a second bottom TCP (BTCP) module located at said selected server computer (Spec., p. 39, ln. 24-25);

changing source addresses of said response packets to a first IP address of said first server computer (Spec., p. 39, ln. 25-26);

updating sequence numbers and TCP checksum in said response packets to reflect said first TCP state of said first server computer (Spec., p. 39, ln. 27- p. 40, ln. 1); and

sending said response packets to said client (Spec., p. 40, ln. 1-2).

Dependent claim 10 recites the method as described in Claim 1, comprising the further steps of:

monitoring TCP/IP control traffic for said communication session at said second BTCP module (Spec., p. 32, ln. 24-26);

understanding when said communication session is closed at said second server computer (Spec., p. 32, ln. 24-26);

sending a termination message to said first server computer over said control channel (Spec., p. 32, ln. 24-26);

terminating said TCP/IP communication session at said first server computer by terminating a forwarding mode at said first BTCP module (Spec., p. 32, ln. 24-26); and

freeing data resources associated with said communication session at said first server computer (Spec., p. 32, ln. 24-26).

Dependent claim 12 recites the method as described in Claim 11 , wherein said step a) comprises the steps of:

receiving a packet from said client at said first BTCP module (Fig. 10:1010; Spec., p. 33, ln. 9-12);

sending said SYN packet upstream to a first TCP module located above said first BTCP module in a first operating system of said first server computer (Fig. 10:1020; Spec., p. 33, ln. 13-14);

receiving a first SYN/ACK packet from said first TCP module (Fig. 10:1030; Spec., p. 33, ln. 16-18);

parsing said first initial TCP state from said first SYN/ACK packet, including a first initial sequence number for said first TCP module associated with, said TCP/IP communication session (Fig. 10:1040; Spec., p. 33, ln. 18-20);

sending said SYN/ACK packet to said client (Fig. 10:1050; Spec., p. 33, ln. 20-22);

receiving an ACK packet from said client at said first BTCP module (Fig. 10:1080; Spec., p. 33, ln. 26-29);

sending said ACK packet to said first TCP module (Fig. 10:1070; Spec, p. 33, ln. 25-26);

storing said SYN, ACK and said web request at said first server computer (Fig. 10:1090; Spec., p. 34, ln. 1-5).

Dependent claim 13 recites the method as described in Claim 11 , wherein said step e) comprises the steps of:

sending a handoff request packet from said first BTCP module to said second BTCP module over said control channel (Spec., p. 35, ln. 7-14);

including said SYN packet and said ACK packet in said handoff request packet (Spec., p. 35, ln. 16-18);

changing a first destination IP address of said SYN packet to a second IP address of said selected server computer, at said second BTCP module (Spec., p. 37, ln. 5-8);

sending said SYN packet to said second TCP module (Spec., p. 37, ln. 8-10);

receiving a second SYN/ACK packet at said second BTCP module;

parsing said second initial TCP state from said second SYN/ACK packet, including a second initial sequence number, for said second TCP module, that is associated with said TCP/IP communication session (Spec., p. 37, ln. 12-17);

changing a second destination IP address of said ACK packet to said second IP address, at said second BTCP module (Spec., p. 37, ln. 5-8);

updating said ACK packet to reflect said second TCP state of said selected server computer in said communication session (Spec., p. 37, ln. 19-22);

sending said ACK packet that is updated to said second TCP module (Spec., p. 37, ln. 24-26); and

sending a handoff acknowledgment message to said first BTCP module (Spec., p. 37, ln. 27- p. 38, ln. 2).

Dependent claim 16 recites the method as described in Claim 13, comprising the further steps of:

> intercepting a connection indication message sent from said first TCP module to an application layer above said first TCP module at a first upper TCP (UTCP) module, said connection indication message sent by said first TCP module upon establishing said communication session (Spec., p. 10, ln. 2-5; p. 34, ln. 10-11); and
>
> holding said connection indication message at said first UTCP module (Spec., p. 10, ln. 2-5; p. 34, ln. 10-11).

Dependent claim 17 recites the method as described in Claim 16, wherein step h) comprises the further steps of:

> sending a reset packet from said first BTCP module upon receiving said handoff acknowledgment packet to said first TCP module (Spec., p. 35, ln. 22-25);
>
> discarding said connection indication message at said first UTCP module (Spec., p. 36, ln. 1-4);
>
> receiving incoming data packets from said client at said first BTCP module (Spec., p. 38, ln. 19-21);
>
> changing said destination addresses of said incoming data packets to said second IP address (Spec., p. 38, ln. 25-27);
>
> updating sequence numbers and TCP checksum in said data packets to reflect said second TCP state of said selected server computer (Spec., p. 38, ln. 27-29); and
>
> forwarding said data packets to said selected server computer (Spec., p. 38, ln. 29 – p. 39, ln. 2).

Dependent claim 18 recites the method as described in Claim 11, wherein step k) comprises the steps of:

intercepting outgoing response packets from said selected server computer at said second BTCP module (Spec., p. 39, ln. 24-25);

changing source addresses of said response packets to said first IP address (Spec., p. 39, ln. 25-26);

updating sequence numbers and TCP checksum in said response packets to reflect said first TCP state of said first server computer (Spec., p. 39, ln. 27- p. 40, ln. 1); and

sending said updated-response packets to said client (Spec., p. 40, ln. 1-2).

Dependent claim 19 recites the method as described in Claim 11, wherein step l) comprises the steps of:

monitoring TCP/IP control traffic for said communications session at said second BTCP module (Spec., p. 32, ln. 24-26);

understanding when said communications session is closed at said second server computer (Spec., p. 32, ln. 24-26);

sending a termination message to said first server computer over said control channel (Spec., p. 32, ln. 24-26);

terminating a forwarding mode at said first BTCP module (Spec., p. 32, ln. 24-26); and

freeing data resources associated with said communication session at said first sender computer (Spec., p. 32, ln. 24-26).

Dependent claim 20 recites the method as described in Claim 16, comprising the further steps of:

> sending notification from said first BTCP module to said first UTCP module to release said connection indication message, if said selected server computer is said first server computer (Spec., p. 40, ln. 15-20); and
>
> sending incoming data packets, including said web request, from said client, received at said first BTCP module, upstream (Spec., p. 40, ln. 20-26).

Dependent claim 21 recites the method as described in Claim 11, wherein each of said plurality of server computers is constructed similarly including BTCP modules located downstream from TCP modules, and UTCP modules located upstream from TCP modules (Fig. 5C; Spec., p. 29, ln. 27 – p. 30, ln. 6).

Dependent claim 23 recites the method as described in Claim 22, wherein said control channel allows for communication between all UTCP modules (Spec., p. 30, ln. 28 – p. 31, ln. 3).

Dependent claim 27 recites the server computer as described in Claim 26, wherein said method comprises the steps of

a) establishing a TCP/IP communication session between a client computer and said server computer, said first node, said server computer part of a plurality of server computers forming said cluster network containing information, said communication session established for the transfer of data contained within said information (Spec., p. 22, ln. 22 – p. 23, ln. 19);

b) receiving a web request associated with said TCP/IP communication session at a first BTCP module at said server computer (Spec., p. 34, ln. 18-20);

c) examining content of said web request (Spec., p. 34, ln. 18-20);

d) determining which of said plurality of server computers, a selected server computer, can best process said web request, based on said content (Spec., p. 10, ln. 7-16);

e) handing off said communication session to said selected server computer from said server computer over a persistent control channel, if said selected sender computer is not said server computer (Spec., p. 35, ln. 7-14); and

f) migrating a first TCP state of said server computer to said selected server computer, and sending a second TCP state of said selected server computer to said server computer over said control channel (Spec., p. 11, ln. 8 – pg. 12, ln. 13).

Dependent claim 28 recites the sender computer as described in Claim 27, wherein step a) of said method comprises the steps of:

receiving a SYN packet from said client at said BTCP module (Spec., p. 33, ln. 9-12);

sending said SYN packet upstream to said TCP module (Spec., p. 33, ln. 13-14);

receiving a first SYN/ACK packet from said TCP module (Spec., p. 33, ln. 16-18);

parsing a first initial TCP state from said first SYN/ACK packet, including a first initial sequence number for said TCP module associated with said TCP/IP communication session (Spec., p. 33, ln. 18-20);

sending said SYN/ACK packet to said client (Spec., p. 33, ln. 20-22);

receiving an ACK packet from said client at said BTCP module (Spec., p. 33, ln. 24-25);

sending said ACK packet to said TCP module (Spec., p. 33, ln. 25-26);

storing said SYN, ACK at said server computer (Spec., p. 34, ln. 1-5).

Dependent claim 29 recites the server computer as described in Claim 28, wherein said method comprises the steps of:

sending a handoff request packet from said BTCP module to a second BTCP module over said control channel, said second BTCP module located below a second TCP module in a second operating system at said selected server computer (Spec., p. 35, ln. 7-14);

including said SYN packet and said ACK packet in said handoff request (Spec., p. 35, ln. 16-18);

receiving a handoff acknowledgement message at said BTCP module from said second BTCP module (Spec., p. 37, ln. 27 – p. 38, ln. 2).

Dependent claim 30 recites the server computer as described in Claim 29, wherein said step f) of said method comprises the steps of:

> monitoring traffic associated with establishing said TCP/IP communication session in step a), at said BTCP module, to parse a first initial TCP state of said server computer, said first initial TCP state associated with said TCP/IP communication session (Spec., p. 11, ln. 8 – pg. 12, ln. 13); and
>
> migrating said first initial TCP state to said second BTCP module over said control channel by including said first initial TCP state in said handoff request, said first initial TCP state including a first sequence number, such that said second BTCP module can calculate said first TCP state for said server computer in said TCP/IP communication session (Spec., p. 11, ln. 8 – pg. 12, ln. 13).

Dependent claim 31 recites the server computer as described in Claim 29, wherein said method comprises the further steps of:

> intercepting a connection indication message sent from said first TCP nodule to an application layer above said first TCP module at a first upper TCP (UTCP) module, said connection indication message sent by said first TCP module upon establishing said communication session (Spec., p. 10, ln. 2-5; p. 34, ln. 10-11); and
>
> holding said connection indication message at said first UTCP module (Spec., p. 10, ln. 2-5; p. 34, ln. 10-11).

Dependent claim 32 recites the computer system as described in Claim 31, wherein said method comprises the further steps of:

> sending a reset packet from said first BTCP module upon receiving said handoff acknowledgment packet to said first TCP module (Spec., p. 35, ln. 22-25);
>
> discarding said connection indication message at said first UTCP module (Spec., p. 36, ln. 1-4);
>
> receiving incoming data packets from said client at said first BTCP module (Spec., p. 38, ln. 19-21);
>
> changing said destination addresses of said incoming data packets to said second IP address (Spec., p. 38, ln. 25-27);
>
> updating sequence numbers and TCP checksum in said data packets to reflect said second TCP state of said selected server computer (Spec., p. 38, ln. 27-29); and
>
> forwarding said data packets to said selected server computer (Spec., p. 38, ln. 29 – p. 39, ln. 2).

Dependent claim 33 recites the server computer as described in Claim 31, said method comprising the further steps of:

> sending notification from said BTCP module to said UTCP module to release said connection indication message, if said selected server computer is said server computer (Spec., p. 40, ln. 15-20);
>
> sending incoming data packets, including said web request, from said client, received at said first BTCP module, upstream (Spec., p. 40, ln. 20-26).

Dependent claim 34 recites the server computers described in Claim 26, said method comprising the further steps of:

receiving a handoff request from a first BTCP module located at a first server computer within said cluster network over a persistent control channel, said first server computer having established a communication session with a client computer, said communication session established for the transfer of data contained within said server computer, said handoff request including a SYN packet and an ACK packet, said SYN and ACK packet used for establishing said communication session between said client and said first server computer, said ACK packet including a first initial TCP state of said first server computer in said communication session, including a first initial TCP sequence number (Spec., p. 35, ln. 7-14);

changing a first destination IP address of said SYN packet to a second IP address of said server computer, at said BTCP module (Spec., p. 37, ln. 5-8);

sending said SYN packet to said TCP module (Spec., p. 37, ln. 8-10)

receiving a SYN/ACK packet at said second BTCP module;

parsing a second initial TCP state from second SYN/ACK packet, including a second initial sequence number, for said TCP module, said second initial TCP state associated with a second TCP state for said server computer in said TCP/IP communication session (Spec., p. 37, ln. 12-17);

changing a second destination IP address of said ACK packet to said second IP address, at said BTCP module (Spec., p. 37, ln. 5-8);

updating said ACK packet to reflect said second TCP state of said selected server computer in said communication session (Spec., p. 37, ln. 19-22);

sending said ACK packet that is updated to said TCP module (Spec., p. 37, ln. 14-26); and

sending a handoff acknowledgment message to said first BTCP module over said control channel (Spec., p. 37, ln. 27 – p. 38, ln. 2).

Dependent claim 35 recites the server computer as described in Claim 31, wherein said method comprises the further steps of:

> monitoring traffic associated with handing off said TCP/IP communications session to said server computer, at said BTCP module, to parse said second initial TCP state of said server computer, said second initial TCP state associated with said TCP/IP communication session (Spec., p. 11, ln. 8 – p. 12, ln. 13); and
>
> sending said second initial TCP state of said server computer to said first BTCP module by including said second initial TCP state in said handoff acknowledgment, said second initial TCP state including a second initial sequence number, such that said first BTCP module can calculate said second TCP state for said server computer in said TCP/IP communication session (Spec., p. 11, ln. 8 – p. 12, ln. 13).

Dependent claim 36 recites the server computer as described in Claim 34, wherein said method comprises the further steps of:

> intercepting outgoing response packets from said server computer at said second BTCP module (Spec., p. 39, ln. 24-25);
>
> changing source addresses of said response packets to said first IP address (Spec., p. 39, ln. 25-26);
>
> updating sequence numbers and TCP checksum in said response packets to reflect said first TCP state of said first server computer (Spec., p. 39, ln. 27- p. 40, ln. 1); and
>
> sending said response packets to said client (Spec., p. 40, ln. 1-2).

Dependent claim 37 recites the server computer as described in Claim 31, wherein said method comprises the further steps of:

> monitoring TCP/IP control traffic for said communication session at said BTCP module (Spec., p. 32, ln. 24-26);
>
> understanding when said communication session is closed at said server computer (Spec., p. 32, ln. 24-26); and
>
> sending a termination message to said first server computer over said control channel (Spec., p. 32, ln. 24-26).

X. RELATED PROCEEDINGS

Applicant respectfully notes that the following co-pending applications are or have been previously on appeal before the Board, which may have a bearing on the Board's decision in this appeal: 1) Application No. 09/880,632 ("the '632 Application"); 2) Application No. 10/146,988 ("the '988 Application); and 3) Application No. 10/146,967 ("the '967 Application").

An appeal was filed for the '632 Application with an Appeal Brief being filed August 20, 2007. An Examiner's Answer was mailed December 13, 2007. As of this time, no decision has been rendered by the Board for the appeal of the '632 Application.

An appeal was filed for the '967 Application with an Appeal Brief being filed February 17, 2009. An Examiner's Answer was mailed April 29, 2009. As of this time, no decision has been rendered by the Board for the appeal of the '967 Application.

An appeal was filed for the '988 Application with an Appeal Brief being filed June 9, 2006. An Examiner's Answer was mailed April 18, 2007, and a Reply Brief was filed June 18, 2007. As of this time, no decision has been rendered by the Board for the appeal of the '988 Application.